**Machine Learning Algorithms(9) — Ensemble techniques (Bagging —Random Forest Classifier and Regression )**

[[Kasun Dissanayake](https://kasunprageethdissanayake.medium.com/?source=post_page-----5d3747c7a953--------------------------------)](https://kasunprageethdissanayake.medium.com/?source=post_page-----5d3747c7a953--------------------------------)

[[Towards Dev](https://towardsdev.com/?source=post_page-----5d3747c7a953--------------------------------)](https://towardsdev.com/?source=post_page-----5d3747c7a953--------------------------------)

[Kasun Dissanayake](https://kasunprageethdissanayake.medium.com/?source=post_page-----5d3747c7a953--------------------------------)

·

[Follow](https://medium.com/m/signin?actionUrl=https%3A%2F%2Fmedium.com%2F_%2Fsubscribe%2Fuser%2F4a3ca0851dfe&operation=register&redirect=https%3A%2F%2Ftowardsdev.com%2Fmachine-learning-algorithms-9-ensemble-techniques-bagging-random-forest-classifier-and-5d3747c7a953&user=Kasun+Dissanayake&userId=4a3ca0851dfe&source=post_page-4a3ca0851dfe----5d3747c7a953---------------------post_header-----------)

Published in

[Towards Dev](https://towardsdev.com/?source=post_page-----5d3747c7a953--------------------------------)

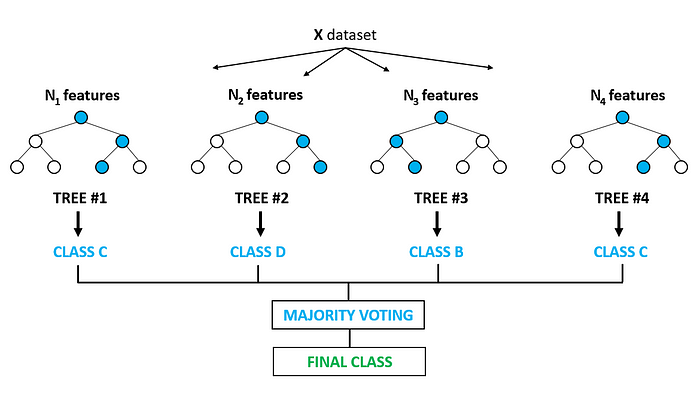
·

8 min read

·

Nov 27

350



Inthis article, I am going to explain to you Ensemble techniques and one of the famous Ensemble techniques which belongs to the **Bagging technique** called **Random Forest Classifier and Regression**.

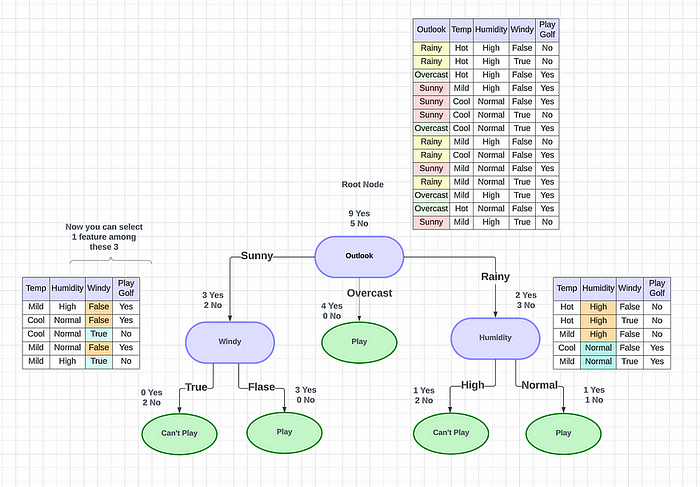
Ensemble techniques are machine learning techniques that combine multiple base modules and models to create an optimal predictive model. To comprehend this definition better, we need to take a step back and consider the ultimate objective of machine learning and model building. Once we have a clear idea of that, we can delve into specific examples and the reasons why ensemble models are preferred. In the previous article, we learned about Decision Trees.

**[Machine Learning Algorithms(8) — Decision Tree Algorithm](https://towardsdev.com/machine-learning-algorithms-8-decision-tree-algorithm-533b6926ddbb?source=post_page-----5d3747c7a953--------------------------------" \t "_blank)**

[In this article, I will focus on discussing the purpose of decision trees. A decision tree is one of the most powerful…](https://towardsdev.com/machine-learning-algorithms-8-decision-tree-algorithm-533b6926ddbb?source=post_page-----5d3747c7a953--------------------------------" \t "_blank)

[towardsdev.com](https://towardsdev.com/machine-learning-algorithms-8-decision-tree-algorithm-533b6926ddbb?source=post_page-----5d3747c7a953--------------------------------" \t "_blank)

This article discusses using a decision tree to **determine whether individuals should play golf outside in certain weather conditions**. The tree takes into account various weather factors, and based on each factor, the tree either decides or asks another question. For example, if it is **overcast**, the decision is to **play outside**, but if it is Sunny, Rainy, or Windy the tree asks further questions before deciding whether or not to play.



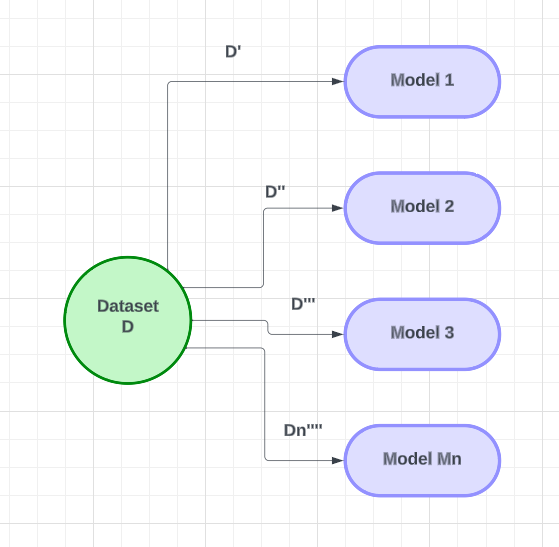
To create a decision tree, we must consider what features we will use to make our decisions and what threshold we will use to classify each question as a **yes or no answer.** We can continue to add questions until we define the yes and no classes. **But what happened if we wanted to ask ourselves if we had friends to play with?** If we have friends, we will play every time. If not, we might continue to ask ourselves questions about the weather. By adding an additional question, we hope to greater define the Yes and No classes. But how we can do this?

Here comes Ensemble Techniques into the picture. Using ensemble methods allows us to take a sample of decision trees into account, determine which features to use at each split, and make a final predictor based on the aggregation results of the sample decision trees. This approach is more reliable than relying on just one decision tree to make our final decision. In Essemble techniques, there are 2 techniques.

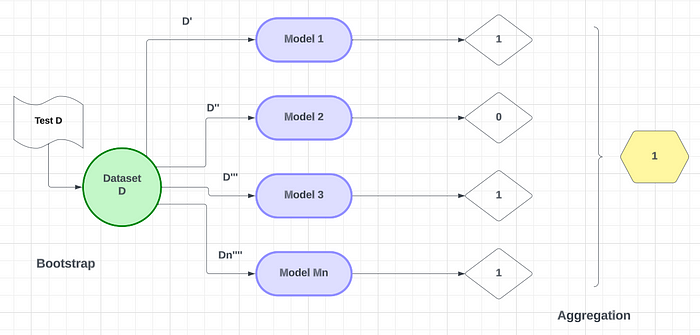
* **Bagging**technique
* **Boosting**technique

**How do Bagging techniques work?**

Bagging also called **bootstrap aggregation**, works by training a dataset with multiple models to obtain a more accurate output. This topic is particularly important as many companies are now using these techniques in their data analysis.



Suppose we have a particular problem statement and we have a dataset called D. In this particular dataset, we will create several base models (M1, M2, M3 …… Mn) and use them to create multiple-based learners. For each model, we will provide a sample of the **dataset(D’)** that is less than the number of records(**n**) in the dataset to the **M1 model**. We will use row sampling with the replacement for each model to provide the data. The same process will be repeated for the next models, where we will resample the records and provide them to the model. Each model will have a different set of data. After providing the data to the models, they will be trained on the data. This process will be repeated for all models in the group.



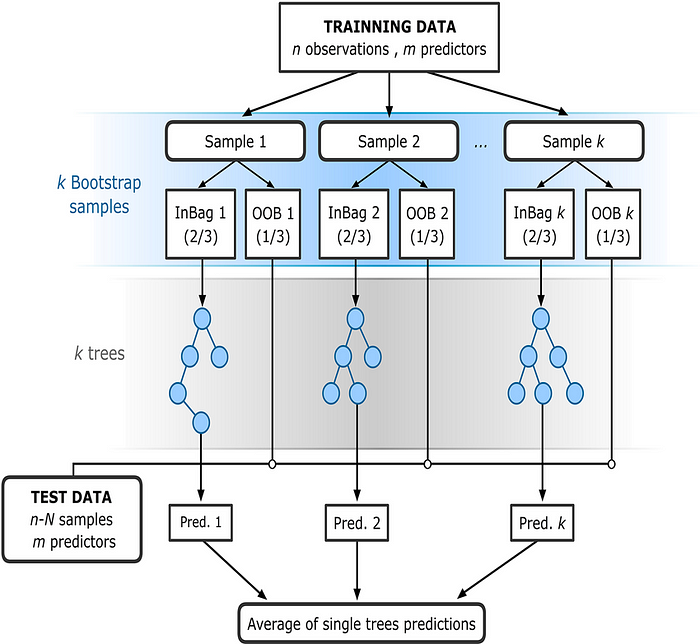
After completing the training, We will use new data from the test dataset to make predictions. **For binary classification**, We will send “**Test D”** data to **Model 1**. If Model 1 **outputs 1, Model 2 outputs 0, Model 3 outputs 1, and Model Mn outputs 1, w**e will use a **voting classifier**to combine the outputs of the models for the test data(**It’s 1**). **The output with the majority of the votes will be considered**. By using row sampling with replacement and a voting classifier, We will combine the outputs of the models for the final result. This is how bagging techniques work.

*Note: For the Regression problem, we can take the output mean as the final result.*

In the bagging technique, we utilize two algorithms,

* **Random Forest Classifier**
* **Random Forest Regression**

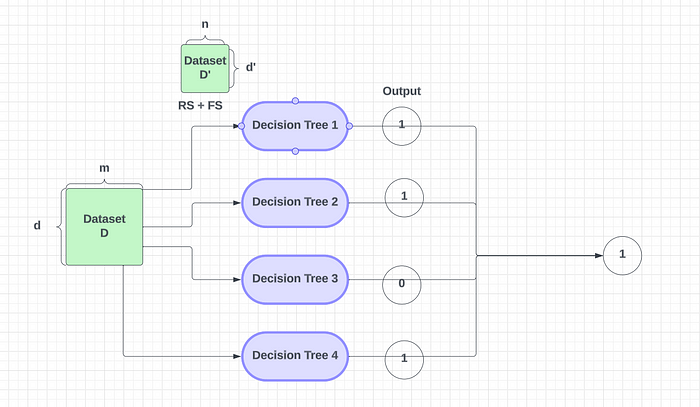
**Random Forest Classifier and Random Forest Regression**



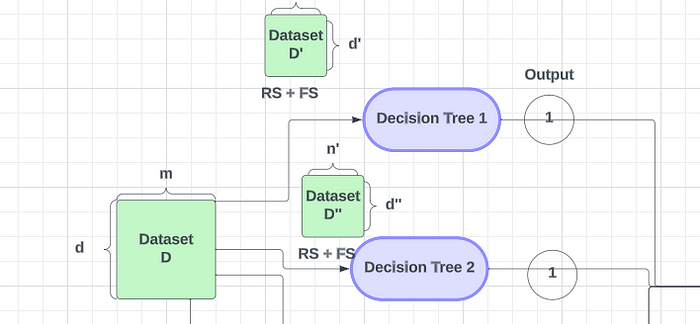
The bootstrapping **Random Forest**algorithm combines ensemble learning methods with the decision tree framework to create multiple randomly drawn decision trees from the data, averaging the results to output a new result that often leads to strong predictions/classifications.

Let me show you some examples to help you understand how random works with a dataset. In **bagging**, we use multiple base learner models, such as Decision Tree 1, Decision Tree 2, and Decision Tree 3…..Decision Tree Mn(We will take 4 Models to explain here). In random forests, **we use decision trees for designing these models.** We sample some rows and columns from the dataset given. We use **row sampling with replacement**, which means we take some rows(**m**) from the dataset and pick some columns(**d**) as **feature samples**. This is how we use bagging to select a subset of rows for our decision tree. The number of records in the decision tree will always be less than the records in the dataset.

Number of Records : m\*d  
Number of Rows : d  
Number of features(columns) : m  
Number of rows selected for row sampling : d'  
Number of features for feature sampling : n  
  
d' < d  
n < m



Here We have a small number of records, so We designate some of them to be used for training. Then, We take a sample of those records and give them to the **first decision tree**. The same process is repeated for the **second decision tree** but with replacement sampling. When we sample with replacement, not all records are repeated — instead, a new sample is taken and given to the second decision tree. Some records and features may get repeated during this process, but many records are changed. This row and feature sampling process is repeated for each decision tree, with a different set of features being used each time.



Row Sampling (RS)+ Feature Sampling (FS)

After training the decision tree on the given data, it can accurately predict outcomes for new test data. **In a binary classification problem**, if the decision tree gives a positive(1) output, we can assume it is positive(1). To make a final prediction, we use the majority vote from the models. For example, if Model 1, Model 2, and Model 4 assume the output is 1, we assume it is positive.

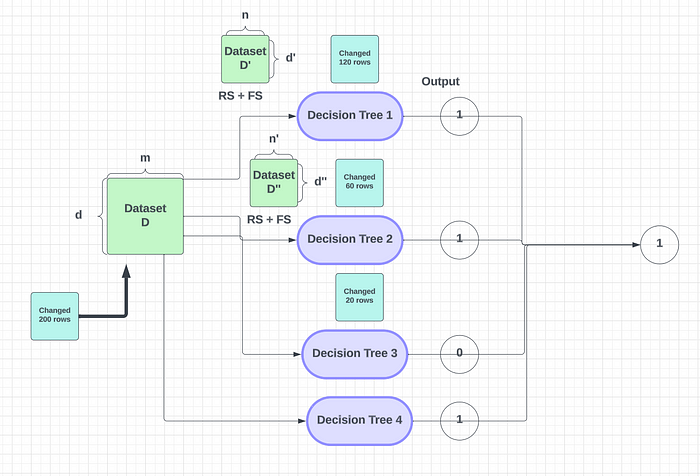
When we use multiple decision trees, we need to consider two properties.

* **Low Bias**
* **High Variance**

If we create a decision tree to its complete depth, it will have low bias and high variance and it will get properly trained to our training dataset. So the training error will be very low.

A high variance occurs when a decision tree **produces a large number of errors due to new test data**. This happens when the decision tree is created to its complete depth and is known as **overfitting**. In random forests, multiple decision trees are used, each with high variance. However, when these decision trees are combined using majority vote, the **high variance is converted into low variance**. This is achieved by using row and feature sampling in datasets. By combining the outputs of multiple decision trees, the high variance is reduced.

**If We have 1000 records and We change 200 records, how will it impact the output?**



Impact on the output

We are currently conducting row and feature sampling for every decision tree change of 200 records. This ensures that the 200 records are properly split between the decision trees. Some of the records will go to decision tree 2 or 1, but this change will **not significantly impact a decision tree’s accuracy or output**. This is due to the **high variance** property of Random Forest, which works well for most machine learning use cases. If this is a regression problem, the decision tree will give a continuous value, and we can either take the mean of all outputs or the median of a particular output.

In Random Forest, the median of a particular output depends on the output’s distribution and the decision tree’s structure. Typically, random forest works by **finding the average of the output across all decision trees**. However, to reduce variance we use multiple decision trees, row sampling, and feature sampling.

Random Forest has both classifiers and regression. The only difference between them is that classifiers use a majority vote, while **regressors find the mean or median of the output** of all decision trees. By adjusting the hyperparameters, such as the number of decision trees, you can optimize Random Forest’s performance.

**Is Nomalizaion(scaling) required in Random Forest or Decision Trees?**

**Nope!**

* The nature of RF is such that convergence and numerical precision issues, which can sometimes trip up the algorithms used in logistic and linear regression, as well as neural networks, aren’t so important. Because of this, you don’t need to transform variables to a common scale as you might with a NN.
* You don’t get any analog of a regression coefficient, which measures the relationship between each predictor variable and the response. Because of this, you also don’t need to consider how to interpret such coefficients which is something that is affected by variable measurement scales.

This is all about the Random Forest Classifier and Random Forest Regression in the Bagging Technique. In the next article, we will learn about Boosting techniques.